FOLC-SR : souple et solide à la fois

Marc-Olivier Buob, Bruno Decraene, Steve Uhlig, Fabien Mathieu

FOLC-SR: souple et solide à la fois

Marc-Olivier Buob[1] †, Bruno Decraene[2], Steve Uhlig[3] et Fabien Mathieu[1]

[1]*Nokia Bell Labs, France*
[2]*Orange Labs, France*
[3]*Queen Mary University of London, United Kingdom*

De nos jours, le routage interne des réseaux cœur est assuré par un protocole à état de liens, appelé IGP (Interior Gateway Protocol). Chaque routeur réapprend en permanence la topologie du réseau IGP, calcule ses plus courts chemins vers chaque destination, et en déduit comment peupler sa table de routage. Cependant, lorsqu'un changement de topologie survient, l'absence de coordination entre les routeurs peut provoquer des boucles de routage transitoires, appelées micro-boucles. Ces dernières engendrent du délai, de la gigue et des pertes de paquets.

Pour résoudre ce problème, nous proposons un mécanisme appelé FOLC-SR (*Fast Optimal Loop-free Convergence over Segment Routing*). FOLC-SR protège rapidement le réseau pour n'importe quel(s) changement(s) de topologie. Pour cela, chaque chemin à risque est décomposé en sous-chemins sûrs d'où sont déduites des règles Segment Routing. Afin de satisfaire les contraintes matérielles des routeurs, FOLC-SR calcule des règles impliquant aussi peu de points de passage que possible. Nous illustrons l'efficacité et le gain apporté par FOLC-SR au travers de simulations conduites sur des topologies réelles.

Mots-clefs : IGP, micro-boucles, algorithme de Dijkstra

1 Introduction

L'Internet est formé d'un ensemble de réseaux IP dont le routage interne est généralement assuré par un protocole *IGP (Interior Gateway Protocol)* à état de liens. Le protocole IGP permet à chaque routeur de cartographier son réseau sous la forme d'un graphe pondéré orienté, de calculer ses plus courts chemins vers chaque destination, et d'enregistrer dans sa table de routage FIB (Forwarding Information Base) quelle(s) interface(s) utiliser pour joindre une destination. Cela permet de réaliser un routage saut-par-saut.

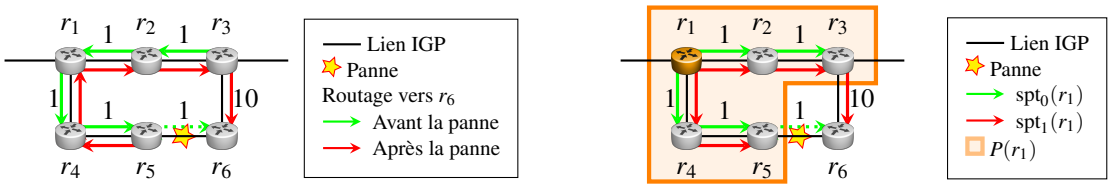
Lorsqu'un routeur découvre un changement de topologie, par exemple à la suite d'une panne, il recalcule ses plus courts chemins et corrige sa FIB. Le calcul est en lui-même très rapide, de l'ordre de quelques millisecondes, mais l'écriture du correctif dans la FIB est sensiblement plus lente, jusqu'à plusieurs centaines de millisecondes pour un grand réseau. Comme les routeurs ne se synchronisent pas pour mettre à jour leur FIB, rien ne garantit que le routage est cohérent de bout en bout pendant les différentes mises à jour. Des boucles de routage transitoires, appelées *micro-boucles*, peuvent alors se former [FB05].

La FIGURE 1a donne un exemple de micro-boucle : on suppose que le routeur r_1 souhaite envoyer un paquet vers le routeur r_6 alors qu'une panne affecte le lien (r_5, r_6). Si sa FIB tient compte de la panne, alors r_1 transmet le paquet vers r_2 (flèche rouge). Si r_2 n'a pas encore corrigé son routage vers la destination r_6, alors il renvoie le paquet à r_1 (flèche verte) : le paquet se trouve piégé dans une micro-boucle.

Les micro-boucles ont un impact néfaste sur le réseau : en plus d'entraîner des pertes de paquets pour la ou les destinations impactées (ici r_6), elles peuvent saturer les liens où les paquets sont piégés, engendrant du délai et de la gigue pour l'ensemble du trafic qui les traverse.

Plusieurs approches ont été proposées pour tenter d'enrayer ce problème. Une première stratégie consiste à contrôler l'ordre dans lequel les routeurs se mettent à jour [FB05, LBU09]. Cela évite de former une micro-boucle lorsque le routeur corrige sa FIB. Chaque correction améliore la situation, mais les destinations qui n'ont pas encore été corrigées peuvent être inatteignables. Une seconde stratégie consiste à installer des règles temporaires plus strictes que le routage saut-par-saut induit par l'IGP, afin d'éviter qu'un routeur non

†Une partie de ces travaux a été effectuée au sein du LINCS (https://www.lincs.fr).



(a) La panne (r_5, r_6) est source de micro-boucles potentielles.

(b) P-space de r_1. Seule la destination r_6 est risquée.

FIGURE 1: Un scénario de panne. Les plus courts chemins dépendent des poids indiqués.

mis à jour ne dévie le trafic. Ces règles peuvent être calculées a priori, comme le proposent les solutions dites de *FastReRoute (FRR)*, ou en réaction aux changements de topologie, et s'appuient sur des tunnels ou du *Segment Routing* (voir plus bas). Les solutions de FRR permettent de réagir plus rapidement, mais ne garantissent pas forcément un routage cohérent de bout en bout. De plus, comme les changements potentiels de topologie sont trop nombreux pour être tous anticipés, une solution de FRR n'en couvre qu'une petite fraction (e.g. les pannes simples de ses liens ou de ses voisins). Certaines approches (NearSide, FarSide) [BFSS15] proposent de réagir en établissant dynamiquement un tunnel. Malheureusement, ces deux solutions ne considèrent que les pannes simples de lien.

Contributions : Nous proposons FOLC-SR, un mécanisme de protection local à chaque routeur qui s'appuie sur Segment Routing (SR). FOLC-SR caractérise les destinations à protéger et permet de calculer des règles SR optimales. Sa souplesse permet de protéger le réseau contre les micro-boucles face à des changements arbitraires de topologie. On aboutit à un routage robuste de bout en bout et conforme à la nouvelle topologie. Nos simulations montrent que pour un coût raisonnable, FOLC-SR est efficace et permet de protéger le réseau contre les situations non couvertes par FRR.

2 Segment Routing

Segment Routing (SR) [FNP+15] permet d'indiquer dans un paquet, en plus de sa destination, une séquence de points de passages intermédiaires à traverser (arc ou nœud). C'est une forme de *loose source routing*. L'énorme intérêt de SR réside dans sa flexibilité (n'importe quel chemin peut être potentiellement encodé) et dans l'absence de signalisation à la création d'une règle (celle-ci est directement exploitable). De nos jours, SR est une technologie supportée par la quasi-totalité des équipementiers.

Avec SR, chaque nœud et arc est au préalable identifié par un label global appelé *Segment ID (SID)*, appris grâce à l'IGP. Ces SIDs sont utilisés pour définir des règles SR. On note $L(u)$ le SID associé à un nœud u (label de nœud) et $L(e)$ le SID associé à un arc e (label d'adjacence). Lorsqu'un paquet SR atteint un routeur u, le routeur examine le premier SID inscrit dans le paquet.

— S'il s'agit d'une de ses adjacences, u le retire, puis transmet le paquet via l'interface correspondante.
— S'il s'agit de son propre label de nœud, u le retire, puis traite le SID suivant.
— S'il s'agit du label d'un autre routeur v, u transmet le paquet comme s'il était destiné à v.

3 P-space et destinations risquées

La notion de P-space est au cœur de notre mécanisme : elle permet de déterminer quelles destinations protéger et comment les protéger. Le P-space d'un routeur u, noté $P(u)$, est l'ensemble des destinations qui ne sont pas affectées par les changements détectés par u ‡.

Formellement, soit une rafale de changements de topologie transformant le graphe IGP initial G_0 en $G_1 \dots G_n$. Le cas $n = 1$ correspond à un unique changement, par exemple une panne simple, tandis que le cas $n > 1$ peut modéliser une cascade de pannes survenant pendant la mise à jour des FIBs. Pour tout $i \in \{0, \dots, n\}, s \in V_i, t \in V_i$, $\text{sp}_i(s,t) \subseteq E_i$ désigne l'ensemble des arcs impliqués dans un plus court chemin allant de s à t dans G_i. Si $u \notin V_i$ ou $t \notin V_i$, par convention $\text{sp}_i(u,t) = \emptyset$. Enfin, on note $\text{spt}_i(s) = \cup_{t \in V_i}(\text{sp}_i(s,t))$ le graphe acyclique des plus courts chemins de s dans G_i §. Le P-space d'un nœud u lors du changement $G_0 \dots G_n$ est défini par : $P(u) = \{u\} \cup \{t \in V_0 \cup \dots \cup V_n, \text{sp}_0(u,t) = \dots = \text{sp}_n(u,t)\}$.

‡. Notre définition est plus stricte que celle introduite dans [BFSS15], mais se généralise mieux et permet des calculs plus efficaces.
§. En absence d'égalité, c'est plus simplement l'*arbre* des plus courts chemins issus de s.

Par construction, toute destination dans le P-space est *sûre* (depuis u) car le routage n'est pas impacté. Une destination hors de $P(u)$ est au contraire *risquée*. Si FOLC-SR est activé sur u, il doit protéger chaque destination hors de $P(u)$. La FIGURE 1b illustre cette notion : on voit que $r_6 \notin P(r_1)$, donc r_1 doit protéger la destination r_6 selon le mécanisme que nous allons maintenant décrire.

4 Protection des destinations

Pour protéger une destination depuis un routeur donné, on peut réaliser un routage à la source en énumérant les SIDs de tous les arcs du chemin à suivre. Sur l'exemple décrit en FIGURE 2, pour aller de s à t, cela donne la protection naïve $[L(s,a), L(a,b), L(b,c), L(c,d), L(d,e), L(e,t)]$, qui implique six SIDs.

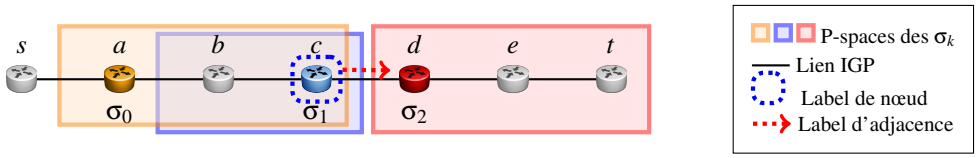


FIGURE 2: La protection calculée FOLC-SR par s pour protéger t est $[L(c), L(c,d)]$.

Cependant, un routeur ne peut écrire au niveau matériel que peu de SIDs (de l'ordre de 4). Cette valeur, appelée MSD (Maximum SID Depth), est souvent insuffisante pour réaliser des protections naïves. Nous proposons donc un algorithme qui minimise le nombre de SIDs utilisés. Soit μ le plus court chemin de s à t à protéger dans la topologie finale G_n. Notre algorithme calcule une séquence de points de passage $(\sigma_0, \ldots, \sigma_k, \ldots)$ suffisante et de cardinalité minimale. On note σ_0 le successeur de s dans μ. On part de ce routeur (au lieu de s) car s devra de toute manière indiquer quelle interface de sortie utiliser dans sa FIB. Cette astuce permet de « gagner » un saut. Étant donné σ_k, le point de passage suivant σ_{k+1} est :

— le nœud de $\mu \cap P(\sigma_k)$ le plus proche de t s'il est distinct de σ_k. On ajoute alors $L(\sigma_{k+1})$ à la protection.
— le successeur de σ_k dans μ sinon. On ajoute alors $L(\sigma_k, \sigma_{k+1})$ à la protection.

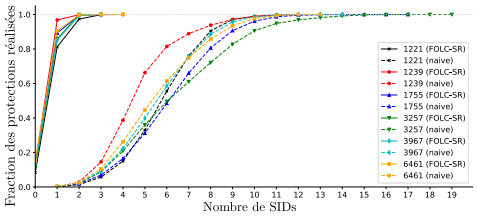
On s'arrête dès que $t \in P(\sigma_k)$. Chaque σ_k engendre implicitement le calcul de $P(\sigma_k)$ par le routeur s.

La FIGURE 2 illustre notre algorithme. Depuis a, on peut atteindre au mieux c sans risque. On ajoute donc $L(c)$ à la protection. $P(c)$ ne permettant pas d'avancer dans μ, on ajoute $(L(c,d)$ à la protection. Comme $t \in P(d)$ on aboutit à la protection $[L(c), L(c,d)]$, qui n'implique plus que 2 SIDs.

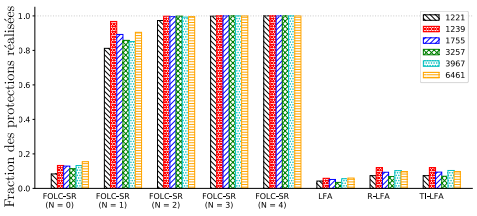
Par définition du P-space, la protection permet à s d'atteindre t de manière sûre. On peut montrer que le nombre de SIDs utilisés est minimal (le cœur de la preuve est que pour tout nœud u de $P(\sigma_k)$, $P(u)$ ne permet jamais de progresser au-delà de σ_{k+1} le long de μ). Le routage engendré est sûr et conforme à μ.

Répéter ce raisonnement pour chaque destination risquée serait très coûteux, mais il est possible de mutualiser les calculs, car chaque destination risquée se protège de la même manière que le dernier point de passage de sa protection. Par exemple, sur la FIGURE 2, d, e et t se protègent de la même façon. Ce raisonnement s'étend naturellement sur $\text{sp}_n(s,t)$, et même sur $\text{spt}_n(s)$. Cela permet, aux calculs de P-space près, de protéger l'ensemble du réseau depuis s en effectuant un seul parcours de $\text{spt}_n(s)$.

5 Simulations



(a) CDF du nombre de SIDs utilisés (naïf vs FOLC-SR).



(b) Comparaison de FOLC-SR et de FRR.

FIGURE 3: Mesure de la fraction des protections réalisées sur les réseaux Rocketfuel pour l'ensemble des triplets *source, destination, panne simple* possibles.

Nous avons testé notre implémentation de FOLC-SR par simulation sur les topologies d'AS du projet Rocketfuel ¶. Nous avons considéré tous les scénarios de panne simple de nœud ou de lien.

La FIGURE 3a illustre l'efficacité de notre algorithme de compression de protection : FOLC-SR est capable de protéger 100% des scénarios de panne en moins de 4 SIDs, ce qui est réalisable au niveau matériel même sur des équipements relativement anciens. À l'inverse, seule une petite fraction des pannes peuvent être protégées de manière naïve en moins de 4 SIDs.

La FIGURE 3b détaille FOLC-SR pour un nombre N de SIDs autorisés variant de 0 à 4, et le compare à LFA, R-LFA [BFSS15] et TI-LFA [LBF+19], trois solutions pro-actives de type FRR, configurées sur chaque routeur pour couvrir les pannes d'équipement adjacent ‖. Ces trois solutions couvrent moins de 20% des scénarios. Pour FOLC-SR, le score obtenu pour $N = 0$ revient à évaluer la fraction de situations résolues juste en corrigeant l'interface de sortie (c'est ce que fait l'IGP). Dès $N = 2$ l'essentiel des cas sont protégés (on peut montrer que 2 SIDs suffisent à gérer toute panne simple de lien sur des graphes symétriques bi-connexes). Pour $N = 4$, l'ensemble des pannes simples (lien et routeur) sont couvertes.

Le temps pris par FOLC-SR dépend directement du nombre de P-spaces à calculer. Hormis pour l'AS1239, une implémentation C++ de FOLC-SR permet de calculer l'ensemble des protections en moins de 5*ms*. Dans l'AS1239, significativement plus important, 75% des protections sont calculées en moins de 25*ms* et le reste, correspondant à des scénarios où l'essentiel de la FIB doit être corrigée, en moins de 200*ms*.

Nous avons observé des résultats similaires sur des échantillons de pannes doubles ou triples.

6 Conclusion

FOLC-SR protège un réseau IGP contre le phénomène de micro-boucle. Lorsqu'un routeur active FOLC-SR, il calcule des règles de protection, basées sur Segment Routing, lui permettant d'atteindre chaque destination via ses nouveaux plus courts chemins. Ces règles traduisent un chemin en une séquence minimale de points de passages, en général suffisamment concise pour être réalisable au niveau matériel. À l'image d'un IGP, FOLC-SR réagit de manière agnostique aux changements de topologie (aucune hypothèse n'est nécessaire sur ce changement). Cela le distingue des solutions pro-actives de type FRR, qui ne peuvent précalculer tous les changements possibles. FOLC-SR est purement local au routeur, et requiert juste que SR (aujourd'hui supporté par tous les grands fabriquants) soit activé sur les autres routeurs. Cette conception permet d'envisager un déploiement progressif sur un réseau opérationnel, la protection augmentant avec le nombre de routeurs implantant FOLC-SR.

Références

[BFSS15] S. Bryant, C. Filsfils, M. Shand, and N. So. RFC 7490 : Remote loop-free alternate (LFA) fast reroute (FRR). April 2015.

[FB05] P. Francois and O. Bonaventure. Avoiding transient loops during IGP convergence in IP networks. In *IEEE INFOCOM*, volume 1, pages 237–247. IEEE, 2005.

[FNP+15] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois. The segment routing architecture. In *IEEE GLOBECOM*, 2015.

[LBF+19] S. Litkowski, A. Bashandy, C. Filsfils, B. Decraene, P. Francois, D. Voyer, F. Clad, and P. Camarillo Garvia. Topology Independent Fast Reroute using Segment Routing. Internet-draft, IETF, 2019.

[LBU09] A. Lambert, M.-O. Buob, and S. Uhlig. Improving internet-wide routing protocols convergence with MRPC timers. In *ACM CoNEX*, pages 325–336, 2009.

[SB10] M. Shand and S. Bryant. RFC 5715 : A framework for loop-free convergence, 2010.

¶. https ://research.cs.washington.edu/networking/rocketfuel/

‖. Les opérateurs se limitent typiquement à ces pannes car les systèmes pro-actifs ne peuvent matériellement considérer qu'un nombre limité de scénarios. Les systèmes réactifs n'ont pas cette limitation.